Abstract of ——Using recipe ingredients to categorize the cuisine

a. Title: Using recipe ingredients to categorize the cuisine
b. Team Members: Fan Hu, JianPing Zhang, Manshan Lin, Xin Tong
c. E-mail: Jianpingzhang2018@u.northwestern.edu, xintong2018@u.northwestern.edu
d. Course: EECS—349 Machine Learning
e. Task:
i. What is the task, and why is it important/interesting?

Different countries and regions have different cooking habits and corresponding recipes. So Our task is to build a classifier to determine the cuisine of a recipe based on its ingredients. The reason why we attach great importance to this task is that this classifier can serve as important reference for customers who go to unfamiliar restaurant and order unfamiliar food. Moreover, such task can cultivate our abilities to apply different machine learning algorithms into specific real world problems and of course, to use effective data-preprocessing techniques to deal with the raw data.

ii. Describe your approach in high-level terms: what kind of learner(s) did you use, what types of features did you use.

We applied different algorithms including decision tree, random forest, logistic regression (L1 & L2), SVM and KNN, adjusted the parameters of each algorithm, and tried two methods to reduce dimension—— eliminating all parents-children structure, and PCA. We used the feature “ingredient” to predict the feature “cuisine”.

iii. Describe the key results (how well your solution performed in no more than a paragraph, along with your key findings, e.g. which learners performed best, which features were most important)

We did works as follows: **1. Feature Engineering.** We build the bag of words and tried TF*IDF Method to deal with our dataset. **2.Model Training.** We applied different algorithms and adjusted the parameters, including decision tree, random forest, logistic regression (L2), SVM and KNN. **3. Reduce Dimension.** We tried to eliminate all parents-children structure, reducing to nearly 2000 dimension. Plus, we also used PCA to reduce dimension. **4. Ensemble Models.** Trying model ensemble to combine all models together in order to make full use of their advantages. All in all, we applied different algorithms including decision tree, random forest, logistic regression (L1 & L2), SVM and KNN, adjusted the parameters of each algorithm, and tried different methods to reduce dimension, respectively get best accuracy of 62.1%, 73.1%, 78.4% & 78.4%, 81.1%, and 51.6%.

f. At least one picture or graph that illustrates your work, with a caption explaining what the figure shows and its significance.

The accuracy of each algorithm based on different feature selections or dimension reducing methods.

Accuracy \ Dataset	using all features	using top 2000 features	using top 3000 features	using pca on 3000 features	eliminating parent struc. on 3000 features
Logistic Regression	78.378%	77.685%	78.258%	78.231%	74.202%
Random Forest	73.079%	72.576%	72.976%	58.286%	73.561%
Decision Tree	62.131%	60.250%	61.002%	39.732%	61.253%
KNN	51.627%				
SVM	81.109%	81.109%	81.109%	69.703%	66.512%

Final Report of EECS 349

——Using recipe ingredients to categorize the cuisine

Data preprocessing

We obtain the original data from kaggle. They provide 39774 training samples and 9944 test samples. Each training sample comes with a unique id, a "cuisine" label referring its category and several ingredients. Basically, cuisine is our training target and ingredients are our inputs.

Feature Engineering

We build the bag of words which is used to count the frequency of the occurrence of each ingredients and we build the ingredients dictionary which regards each ingredient as a vector and represents the occurrence by 0/1 code with the use of one-hot encoding.

The result of the word bags is nearly 6000 ingredients and the occurrence frequency is ranging from 1 to 18000. The result of the ingredients Vector is nearly 6000 features and 40000 records. The attribute value of each feature is 0/1 code. We also applied TF*IDF Method to deal with our dataset, and trained it on SVM model.

Model Training

We applied different algorithms including decision tree, random forest, logistic regression (L1 & L2), SVM and KNN, adjusted the parameters of each algorithm, and tried different methods to reduce dimension and get corresponding accuracy(We submitted our test result to get corresponding accuracy from Kaggle.)

For the decision tree and Random Forest, we obtained the accuracy of 62.1% and 73.1% on the testing set. By calculating the information gain, we got the features' importance list which is an important reference for the later re-processing data.

For the logistic regression, we used one versus rest logistic regression to build one regression model for each cuisine, in order to observe weights and display correlated ingredients for each cuisine. Both L1 and L2 version logistic regression are tried and the accuracy on testing set are both around 78%, while l2 version performs slightly better. So we choose L2 penalty and 1.2 penalty weight. We also use parameters of logistic regression to show each ingredient importance for each cuisine.

For the SVM, we chose One-or-Rest support vector classifier method to train the model. To determine the weight of ingredients, we chose the term frequency–inverse document frequency (TF*IDF) method, fully considering the functions of term frequency, term count, and inverse document frequency. Eventually we got 81% accuracy on the testing set.

For the KNN, by calculating the distance based on the ingredients vector, we can determine the cuisine according to its nearest neighbor. The accuracy is around 51% on the testing set.

Reduce Dimension & Ensemble Models

At first, we tried to use top frequency feature dimension reduction strategy. After our tests on dataset, we found out that reducing the whole dataset to the top 3000 features can largely reduce computation complexity with slightly loss of accuracy.

Then, since there are some ingredients which can be seen as parents and children. (e.g. "tomato" and "green tomato"), we did some text processing work, trying to include all children situation to their parents. So we got another data set without the parent-children structure. We re-trained our model using the data which has eliminated the parent-children structure and deciding which data set to use based on the test accuracy.

What’s more, we also tried PCA dimension reduction strategy to reduce dimension, and retrained our model to identify whether we need to use dimension reduction technique. The detailed output are shown on the following table. We can see that based on our dataset, neither PCA dimension reduction strategy or eliminating the parent-children structure have a significant effect on the accuracy. Only Random Forest and Decision Tree have a slightly improvement on accuracy eliminating the parent-children structure. So We decided not use the two dimension reducing methods. But, this attempt is still necessary and worthwhile.

Also, we tried model ensemble to combine all models together in order to make full use of their advantages.

There are some future work we can do if we have more time. For example, we can eliminate some basic ingredients such as "salt" and add more weights to some unique ingredients such as soy sauce for Asia foods. We believe these processing may bring some interesting results.

Accuracy \ Dataset	using all features	using top 2000 features	using top 3000 features	using pca on 3000 features	eliminating parent struc. on 3000 features
Logistic Regression	78.378%	77.685%	78.258%	78.231%	74.202%
Random Forest	73.079%	72.576%	72.976%	58.286%	73.561%
Decision Tree	62.131%	60.250%	61.002%	39.732%	61.253%
KNN	51.627%				
SVM	81.109%	81.109%	81.109%	69.703%	66.512%

Member Duties

Name	Duties
Manshan Lin	KNN, Status Report, Eliminate Parent-Children Structure, Adaboost, PCA
Xin Tong	Data Pre-processing, SVM, Final Report, PCA
Jianping Zhang	Logistic Regression, Proposal, Leader, PCA
Fan Hu	Decision Tree, RandomForest, Website, PCA